CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Creación de una base de datos para almacenar datos de secuenciación genética y su integración con BLAST

Autor: Carlos Carrillo Sanz

Director: Raúl Alonso Calvo

MADRID, FEBRERO DE 2015

RESUMEN

Este Trabajo de Fin de Grado (TFG) consiste en el diseño y el desarrollo de una base de datos para almacenar datos de secuenciación genética. Además, también será necesario poder utilizar la herramienta BLAST, que está formada por un conjunto de programas para buscar por similitud y alinear secuencias, con los datos que se encuentran almacenados en dicha base de datos.

ABSTRACT

The aim of this Bachelor's Thesis is to design and develop a database to store data of genetic sequences. Furthermore, it will be necessary to use BLAST, which is a suite of programs to search and match similarities sequences into a database.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

1. INTRODUCCIÓN

En este capítulo se hará una breve introducción sobre los orígenes de la bioinformática y su importancia en la sociedad.

Además, se explicará los objetivos del proyecto y la estructura de la memoria.

1.1. Motivación

Los Institutos Nacionales de la Salud (en inglés *National Institutes of Health*, NIH)[1] definen bioinformática como la investigación, el desarrollo o la aplicación de herramientas computacionales para la expansión del uso de datos biológicos, médicos, conductuales o de salud, incluyendo aquellas para adquirir, almacenar, organizar, archivar, analizar o visualizar tales datos. Es decir, es la creación de herramientas computacionales para trabajar con datos biológicos y resolver sus problemas.

También define la biología computacional como el desarrollo y la aplicación de análisis de datos y de métodos teóricos, de modelado matemático y técnicas de simulación computacional para el estudio de sistemas biológicos,

[1] http://www.nih.gov/

conductuales y sociales. Es decir, es el estudio de la biología usando técnicas computacionales.

Aunque el uso del término bioinformática es aún reciente, al igual que la informática, no lo es tanto en su contenido. Por ejemplo, en la Grecia Clásica ya hay constancia de la creación de autómatas como dejó reflejado Herón de Alejandría (siglo I d. C), en su tratado *Autómata*, donde explica la creación de mecanismos que imitaban el movimiento, tales como aves que vuelan y beben o puertas que se abren automáticamente, producido por el movimiento del agua, la gravedad o un sistema de palancas. Un equivalente tecnológico en la actualidad serían los robots.

Gran parte de los esfuerzos de la informática se han dirigido a intentar a emular los procesos de control de la información de los seres vivos, es decir, a tratar de reproducir artificialmente fenómenos biológicos. La informática está presente en varias áreas de conocimiento relacionadas con la biología, como la robótica que intenta emular los procesos motores y funcionales de los seres vivos o la inteligencia artificial, que intenta diseñar entidades capaces de resolver cuestiones por sí mismas utilizando como paradigma la inteligencia humana.

La bioinformática está teniendo un avance muy rápido en la actualidad, muestra de ello son sus principales proyectos como, la secuenciación del genoma, el alineamiento de secuencias, la predicción de genes, el montaje del genoma, la predicción de estructura de proteínas o el modelado de la evolución.

Este proyecto está relacionado con la genómica, el área de la biología que se encarga del estudio del funcionamiento, el contenido, la evolución y el origen de los genes. A diferencia de la genética clásica, que a partir de un

fenotipo busca el gen (o los genes) responsables de dicho fenotipo, la genómica tiene como objetivo predecir la función de los genes a partir de su secuencia.

La genómica es una de las áreas más vanguardistas de la biología, cuyo uso ha tenido un importante auge en los últimos años, sobre todo gracias a los avances en la informática y en las técnicas de análisis de genomas y de secuenciación. Por ejemplo, en 2003, dos años antes de lo previsto, se anunció que se había completado el mayor proyecto de ADN realizado hasta la fecha, el Proyecto Genoma Humano (PGH). En la actualidad, hay importantes servidores de acceso público, como el del Centro Nacional para la Información Biotecnológica[2] (*National Center for Biotechnology Information*, NCBI) que permiten que cualquier usuario pueda acceder a la secuencia completa del genoma de varios organismos[3].

1.2. Objetivos

El objetivo de este proyecto es desarrollar una base de datos en la que se puedan almacenar datos de secuenciación genética. El alumno deberá realizar el diseño de dicha base de datos y una aplicación que implemente las funciones necesarias para:

i. Realizar la carga de datos que sigan en formato VCF.

ii. Permitir la ejecución de algunos comandos de la herramienta BLAST sobre un conjunto de datos.

1.3. Estructura del documento

La memoria del trabajo está organizada de la siguiente forma. Primero hay un capítulo donde se explican los conocimientos de la rama de la biología y de

[2] http://www.ncbi.nlm.nih.gov/
[3] Los genomas se pueden encontrar en su servidor FTP: ftp://ftp.ncbi.nih.gov/genomes/.

la genética para poder entender esta memoria. En el siguiente capítulo se detallan las tecnologías utilizadas en el desarrollo de este proyecto. Después se especifica el diseño y la implementación de la solución propuesta a los objetivos de este proyecto. Seguidamente se presenta un ejemplo de uso. En el capítulo final se recogen las conclusiones y las posibles líneas futuras.

CAPÍTULO 2

CONCEPTOS

2. CONCEPTOS

En el siguiente capítulo se explicarán los conceptos relacionados con la biología y la genética necesarios para el entendimiento de la memoria y del proyecto.

2.1. Los ácidos nucleicos

Los ácidos nucleicos son biomoléculas formadas por átomos de carbono, hidrógeno, oxígeno, nitrógeno y fósforo. En concreto, los ácidos nucleicos son polímeros cuyas subunidades se denominan nucleótidos.

Un nucleótido es una molécula formada por la unión de:

- Una base nitrogenada: es un compuesto orgánico cíclico que contiene dos o más átomos de hidrógeno y de carbono. Hay dos tipos de bases nitrogenadas que forman parte de los ácidos nucleicos de los seres vivos: las bases pirimidínicas y las bases purínicas.
    - Las bases pirimidínicas derivan de la pirimidina, cuya fórmula cíclica viene reflejada en la Figura 1. Son la timina, véase en la Figura 2, la citosina, véase en la Figura 3, y el uracilo, véase en la Figura 4, que están representadas por las

letras T, C y U respectivamente. La pirimidina es un compuesto orgánico que posee un único anillo con dos átomos de hidrógeno y dos átomos de carbono.

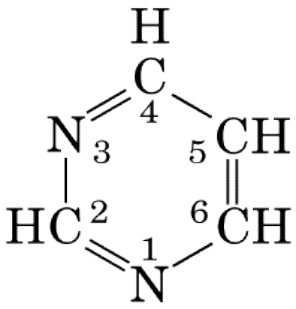

Figura 1: Pirimidina

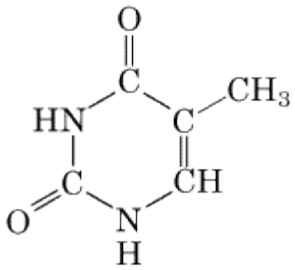

Figura 2: Timina

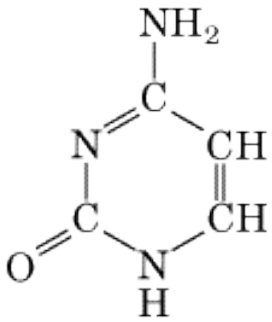

Figura 3: Citosina

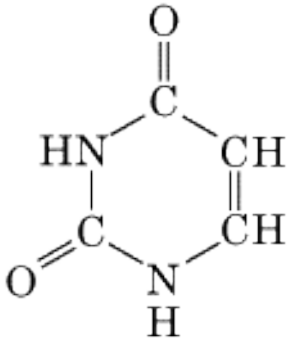

Figura 4: Uracilo

- Las bases purínicas derivan de la purina, cuya fórmula cíclica se puede observar en la Figura 5. Son la adenina, véase la Figura 6, y la guanina, véase la Figura 7, representados por las letras A y G respectivamente. La purina es un compuesto orgánico heterocíclico aromático. Su estructura está compuesta por dos anillos unidos: uno de seis átomos y el otro de cinco. En total estos anillos presentan cuatro átomos de nitrógeno y cinco de carbono.

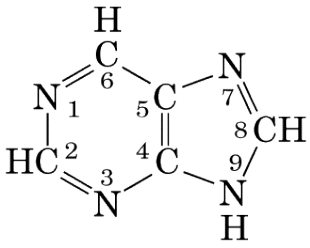

Figura 5: Purina

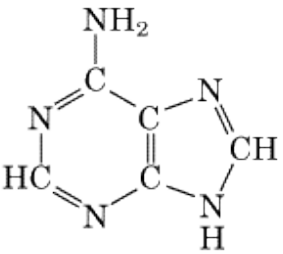

Figura 6: Adenina

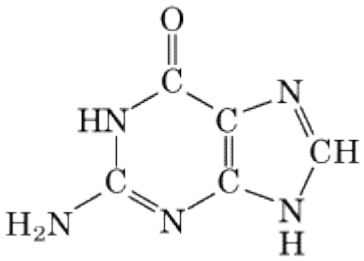

Figura 7: Guanina

- Una pentosa: es un azúcar formado por una cadena de cinco átomos de carbono. Los átomos de carbono se numeran como 1', 2', 3', 4'y 5' para no confundirlos con los átomos de carbono de las bases nitrogenadas. Esta pentosa puede ser la ribosa o la desoxirribosa.

    - Ribosa (β-D-ribofuranosa): es un monosacárido cíclico de cinco átomos de carbono que se encuentra en los nucleótidos del ARN, también llamados ribonucleótidos. Su fórmula ciclada se puede comprobar en la Figura 8.

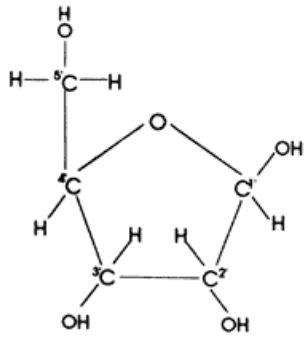

Figura 8: Ribosa

    - Desoxirribosa (β-D-2-desoxirribofuranosa): es un monosacárido cíclico de cinco átomos de carbono que se encuentra en los nucleótidos del ADN, también denominados desoxirribonucleótidos. Su fórmula ciclada se puede observar en la Figura 9. Por tanto, se diferencia de la ribosa en el carbono 2', ya que mientras la desoxirribosa

tiene dos enlaces a sendos átomos de hidrógeno, la ribosa tiene un enlace a un hidrógeno y otro a un grupo hidroxilo.

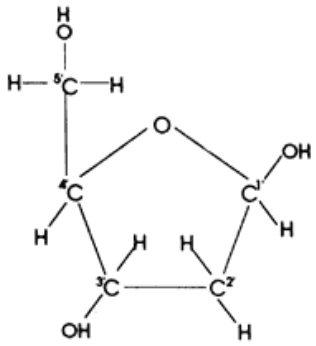

Figura 9: Desoxirribosa

- Un grupo fosfato: el ácido fosfórico (H_3PO_4) se encuentra en los nucleótidos en forma de anión fosfato (PO_4^{3-}).

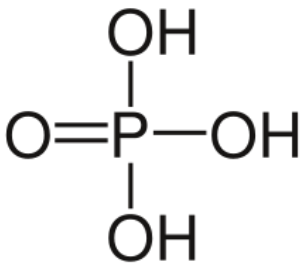

Figura 10: Ácido fosfórico.

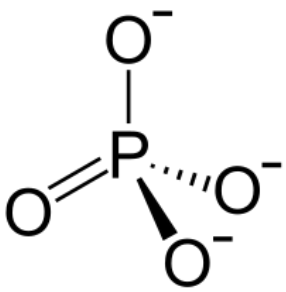

Figura 11: Anión fosfato

La unión covalente entre una base nitrogenada y una pentosa, ya sea ribosa o desoxirribosa, da lugar a un nuevo compuesto llamado nucleósido. Esta unión se lleva a cabo mediante un enlace N-glucosídico que se establece entre el

átomo de carbono carbonílico de la pentosa, C1', y uno de los átomos de nitrógeno de la base, liberándose una molécula de agua. Si la base es pirimidínica entonces se establecerá en el nitrógeno N1 y si es la base es púrica será en el nitrógeno N9.

Los nucleósidos se nombran añadiendo la terminación *-osina* (adenosina, guanosina), si es una base púrica, o *–idina* (citidina, timidina, uridina), si se trata de una base pirimidínica, al nombre de la base nitrogenada. Si la pentosa es la desoxirribosa, se añade el prefijo *desoxi-* (desoxiadenosina, desoxiguanosina, desoxicitidina, desoxitimidina, desoxiuridina).

Los ácidos nucleicos son polinucleótidos formados por la unión de nucleótidos mediante enlaces covalentes de tipo fostodiéster. Estos enlaces se establecen entre sus grupos fosfatos. El fosfato se enlaza por una parte con el carbono 3' de la pentosa de un nucleótido y, por otra, con el carbono 5' de la pentosa siguiente.

Los nucleótidos son los ésteres fosfóricos de los nucleósidos. Es decir, resultan de la esterificación mediante un enlace fosfodiéster establecido entre uno de los grupos hidroxilo de la pentosa de un nucleósido y una molécula de ácido fosfórico en forma de anión fosfato (PO_4^{3-}). Esto le confiere un carácter fuertemente ácido al compuesto. Durante este proceso de esterificación se libera una molécula de agua. Esta unión puede producirse en cualquiera de los grupos hidroxilo libres de la pentosa, pero como regla general tiene lugar en el que ocupa la posición 5'. Se nombran como el nucleósido del que proceden eliminando la *a* final y añadiendo la terminación 5'-fosfato, o bien monofosfato.

Hay dos tipos de ácidos nucleicos: el ADN, o ácido desoxirribonucleico, y el ARN, o ácido ribonucleico. Ambos tipos pueden diferenciarse por su composición química:

- Ácido desoxirribonucleico o ADN: tiene como bases nitrogenadas la adenina, la guanina, la citosina y la timina y como pentosa la desoxirribosa. Se encuentra en el núcleo de las células y forma parte de los cromosomas.

  La estructura de las moléculas de ADN es la de una doble hélice de dos nanómetros de diámetro, formada por dos cadenas helicoidales de nucleótidos enrolladas a lo largo de un eje imaginario común. El enrollamiento es dextrógiro (en el sentido de las agujas del reloj) y plectonémico (para que las dos cadenas se separen es necesario que se desenrollen). Las dos cadenas son antiparalelas, es decir, se disponen paralelas y en sentidos opuestos. Una cadena tiene sentido 5' → 3' (se inicia con un extremo 5' libre y acaba con un extremo 3' libre) y la otra se dispone en sentido 3' → 5'. Las bases nitrogenadas se dirigen hacia el interior de la doble hélice, mientras que las pentosas y los grupos fosfato forman el esqueleto externo. Cada pareja de nucleótidos queda separada de la siguiente por una distancia de 0,34 nanómetros y cada vuelta de la doble hélice comprende 10 pares de nucleótidos, lo que supone una longitud total de 3,4 nanómetros por vuelta.

  La estructura se mantiene estable gracias a los enlaces de hidrógeno que se forman entre los pares de bases nitrogenadas complementarias. La adenina siempre se empareja con la timina mediante dos puentes de hidrógeno y la guanina con la citosina mediante tres puentes de hidrógeno.

  La función del ADN es portar la información hereditaria. La información contenida en el ADN está codificada en forma de

secuencias de bases. Si la secuencia de bases nitrogenadas cambia, la información del ADN también lo hace.

El ADN tiene la capacidad de duplicarse, lo que le permite que su información se herede. La célula utiliza la información contenida en el ADN para elaborar sus propias proteínas, que desempeñarán diversas funciones.

- Ácido ribonucleico o ARN: se forma tomando como molde una cadena de ADN. Por lo general, el ARN es monocatenario, es decir, suele estar formado por una sola cadena de nucleótidos.

  El ARN no suele formar dobles cadenas, salvo en ciertos virus como los *reovirus* (perteneciente a la familia de virus ARN *Reoviridae*).

  La hebra puede plegarse sobre sí misma y formar horquillas (doble hélice con apareamiento de bases complementarias) o bucles (zonas plegadas que no aparean). Tiene como bases nitrogenadas la adenina, la guanina, la citosina y el uracilo y como pentosa la ribosa.

  El ARN se localiza tanto en el núcleo como en el citoplasma celular. Sus principales funciones son copiar la información del ADN para que tenga lugar después la traducción o síntesis de proteínas; portar los aminoácidos específicos hasta los ribosomas y formar los ribosomas. Existen diferentes tipos de ARN que funcionan de manera coordinada:

  - ARN mensajero (ARN_m): se sintetiza a partir del ADN, mediante el proceso de transcripción, y codifica proteínas. Es el responsable de copiar la información del ADN y llevarla hasta los ribosomas con los que colabora en la síntesis de proteínas en el proceso de traducción.

- ARN ribosómico (ARN_r): se une a proteínas básicas para formar los ribosomas.
- ARN de transferencia (ARN_t): formado por pequeñas moléculas encargadas de transportar los aminoácidos a los ribosomas para que se construya la proteína.
- ARN nucleolar (ARN_n): se encuentra unido a diferentes proteínas formando el nucléolo.

En la Figura 12 se puede apreciar las características comentadas anteriormente del ADN y del ARN.

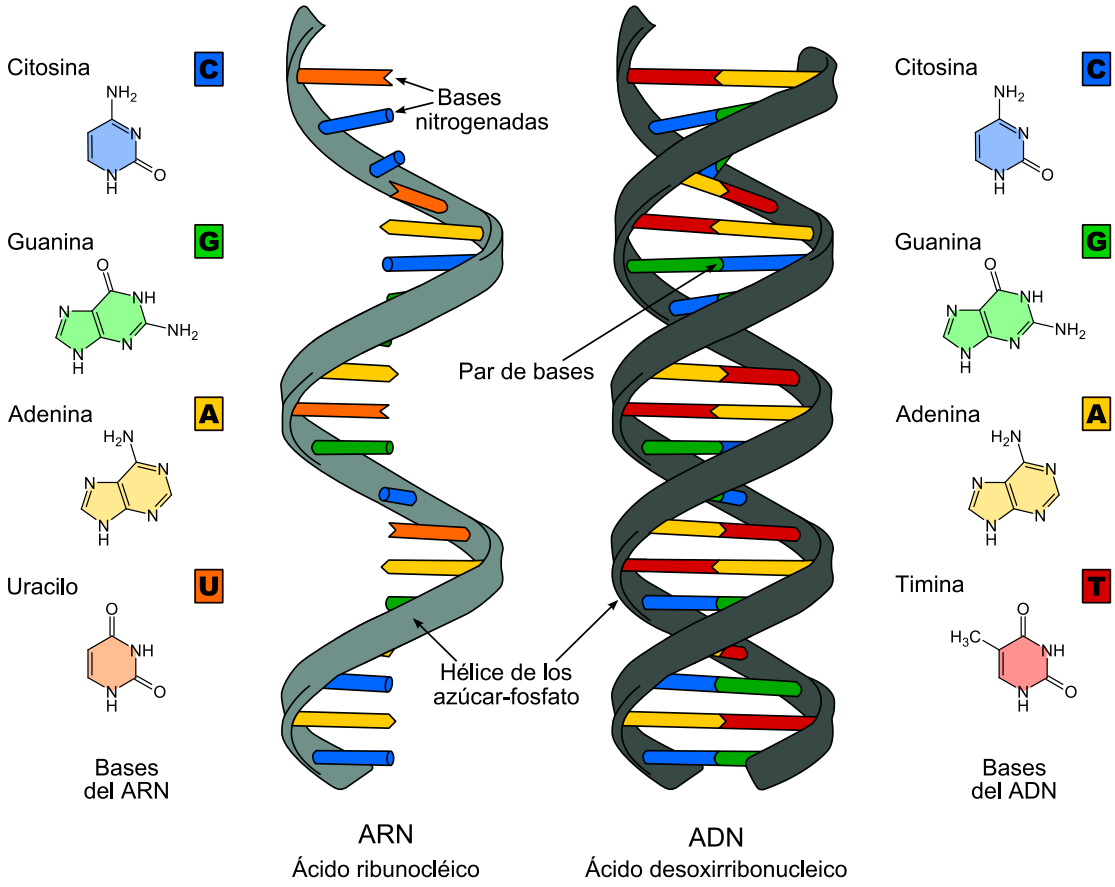

Figura 12: Comparativa entre el ADN y el ARN[4].

2.2.Gen

Un gen es una unidad de información dentro del genoma que contiene todos los elementos necesarios para su expresión de manera regulada. Es decir, es un segmento de ADN, que es una secuencia de nucleótidos (A, G, C y T), que contiene la información necesaria para la síntesis de una macromolécula con función celular específica, habitualmente proteínas pero también ARN_m, ARN_r y ARN_t.

[4]http://upload.wikimedia.org/wikipedia/commons/archive/0/0e/20120808173259%21Difference_DNA_RNA-ES.svg

Los genes de un individuo se encuentran en una serie de estructuras delgadas y filamentosas del cromosoma, denominadas cromátidas, que se localizan dispersas en el citoplasma en los organismos procariontes o en el núcleo en las células eucariotas.

El gen es considerado la unidad de almacenamiento de información genética y unidad de la herencia, pues transmite esa información a la descendencia. Esta información es indispensable para la constitución, el desarrollo y el funcionamiento de los seres vivos.

2.3.Codón

Un codón es un triplete de nucleótidos. Uno o varios codones codifican un aminoácido, y uno o varios aminoácidos se combinan para formar una proteína. Pero por el contrario, un codón no codifica para varios aminoácidos, cada uno lo hace para uno.

2.4.Genoma Humano

Se denomina genoma al conjunto de genes de una especie o de un organismo.

El genoma humano es el conjunto de genes de la especie *Homo sapiens*. Está distribuido en veintitrés pares de cromosomas. De estos veintitrés pares, veintidós son homólogos y la pareja vigesimotercera está formada por el cromosoma XX, en el caso de la hembra, o por el cromosoma XY, en el caso del macho.

El Proyecto Genoma Humano, es un proyecto de investigación científica que surgió con la intención de secuenciar el genoma humano. Secuenciar significa determinar el orden exacto de las bases en un segmento del ADN. En la década de 80, los científicos decidieron secuenciar el genoma humano,

convirtiéndose en uno de los mayores proyectos de investigación de la Historia. En 1988, el Congreso de los Estados Unidos aprobó la financiación del proyecto y eligió a James Dewey Watson, descubridor de la doble hélice de ADN junto a Francis Crick, como el encargado del proyecto. En 1990, se creó un consorcio internacional formado por los siguientes países: Estados Unidos, Reino Unido, Francia, Alemania, Japón y China para la creación del proyecto. En 2003, se declaró su finalización mediante la publicación de una versión terminada de la secuencia del genoma humano. Cuando se publicó que se había terminado de secuenciar el genoma, se refería a que se había completado todo lo posible dentro de los límites de la tecnología del momento, aunque hay pequeños huecos, menos del 1%, que son irrecuperables con cualquier método de secuenciación actual.

El reto de los científicos es entender e identificar los genes del ser humano. Con el conocimiento detallado del genoma humano, se abrirán nuevas vías de avance en la medicina y en la biotecnología. Por ejemplo, se puede conocer si una persona tendrá predisposición a desarrollar una determina enfermedad, como el cáncer de mama o la enfermedad de Alzheimer.

2.5. Mutaciones Genéticas

No existen dos organismos, salvo los gemelos univitelinos, que posean exactamente la misma información genética. Esto se debe a las mutaciones.

Las mutaciones son variaciones que alteran la secuencia de nucleótidos del ADN (el material genético). Pueden aparecer espontáneamente o bien ser inducidas por agentes mutagénicos y son sólo heredables cuando afectan a las células germinales.

Las mutaciones pueden ser perjudiciales, beneficiosas o neutras para el organismo. Se consideran neutras aquellas que no generan ningún beneficio ni perjuicio.

Se denomina alelos a las diferentes alternativas para un mismo carácter. Estas alternativas pueden ser producidas por la mutación de un gen. Por lo tanto, en una misma especie pueden coexistir variantes de cada gen (los alelos) que ocupan la misma posición en el mismo cromosoma.

Las mutaciones se pueden clasificar atendiendo a los siguientes criterios:

- Según el nivel del material genético afectado:
    - Mutaciones génicas, también denominadas mutaciones puntuales porque alteran la secuencia de nucleótidos de un sólo gen. Se producen cuando se modifica la secuencia de nucleótidos del gen, bien porque algún o algunos nucleótidos cambian, o porque cambia su orden. Hay tres tipos de mutaciones génicas:
        - Sustitución de bases: consiste en el cambio de una base por otra. Según la base sustituyente y sustituida, pueden ser:
            - Transiciones: si se sustituye una base púrica por otra púrica o una base pirimidínica por otra pirimidínica.
            - Transversiones: si se sustituye una base púrica por otra pirimidínica, o viceversa.

        Por ejemplo, teniendo las dos siguientes secuencias de nucleótidos: ATGC y ACGC, se ha producido una

sustitución de tipo transición en la posición 2, al cambiarse una timina por una citosina, siendo ambas bases pirimidínicas.

- Inserción: consiste en la inclusión de algún nucleótido en una secuencia. Por ejemplo, teniendo las dos siguientes secuencias de nucleótidos: ATGC y ATAGC, se ha producido una inserción de adenina en la posición 3.

- Deleción: consiste en la pérdida de algún nucleótido en una secuencia. Por ejemplo, teniendo las dos siguientes secuencias de ADN: ATGC y AGC, se ha producido una deleción en la posición 2.

Muchas veces se utiliza la palabra *indel* para referirse a inserción, deleción o a la mutación que incluye inserciones y deleciones. Esto es debido a que se trata de una contracción de las palabras anglosajonas ***in****sertion* y ***del****etion*.

En las mutaciones producidas por la sustitución de bases, sólo un triplete de bases se ve afectado porque estas mutaciones sólo afectan a un único nucleótido. Es posible que este nuevo triplete codifique para el mismo aminoácido que el triplete anterior a la mutación, por lo tanto, esta mutación no afecta al individuo. Esta mutación se conoce como mutación silenciosa. Por el contrario, es más probable que el nuevo triplete codifique para otro nucleótido, pudiendo provocar beneficios en caso de que se produzca una proteína que mejore a la original. Pero también puede provocar perjuicios en caso de que la mutación se produzca en el codón de

terminación, sintetizándose una proteína más corta o más larga que la impida desempeñar su función correctamente. Por último, también puede darse el caso de que no se produzca ninguna consecuencia, por ejemplo en el caso de que la mutación se produzca en una parte de la secuencia no codificante.

En cambio, en las mutaciones producidas por la inserción o deleción, todos los tripletes de bases varían y, en consecuencia, se provoca un corrimiento del marco de lectura, produciéndose aminoácidos distintos.

- Mutaciones cromosómicas: se producen por alteración de la secuencia normal de los fragmentos génicos que componen un cromosoma. Es decir, afectan a la disposición de los genes de un cromosoma, provocando cambios en la estructura de los cromosomas, pero no a la secuencia de nucleótidos de un gen. Pueden afectar al orden de los genes en los cromosomas o a su número. Hay cuatro tipos:

    - Deleción: cuando falta un segmento cromosómico. Si el segmento que falta está en el extremo del cromosoma, entonces se le llama deficiencia.

    - Duplicación: aparece repetido un segmento cromosómico.

    - Translocación: se produce cuando se cambia la localización de un segmento cromosómico.

    - Inversión: el segmento cromosómico ha girado 180º.

- Mutaciones genómicas: alteran el número de cromosomas característico de la especie. Por ejemplo, existe sólo un cromosoma de cada par o hay algún cromosoma de más o de menos.

- Según el tipo de célula afectada. Según este criterio se distinguen:

  - Mutaciones germinales: son las que afectan a las células madres que originan a los gametos o a los propios gametos. Si los gametos intervienen en una fecundación, la mutación se transmitirá a la siguiente generación.

  - Mutaciones somáticas: son las que afectan a células somáticas que forman un individuo. Estas mutaciones afectan al individuo pero no se transmiten a la descendencia, es decir, no son heredables.

2.5.1. Polimorfismo de nucleótido simple

Un polimorfismo de un solo nucleótido (en inglés, *Single Nucleotide Polymorphism*, SNP) es una variación que debe darse al menos en un 1% de la población (si no se llega al 1% no se considera SNP y sí una mutación puntual) en la secuencia de ADN que afecta a un solo nucleótido del genoma. El SNP es la variación más frecuente en el ADN del genoma humano.

2.6. Marcos de lectura

Un marco de lectura es una de las posibles formas de dividir una secuencia de nucleótidos. Por tanto, hay seis posibles marcos de lectura a la hora de leer una doble hélice de ADN, debido a que los codones están formados por tres nucleótidos y a que hay dos cadenas posibles que leer. Por lo tanto, para una

misma cadena de nucleótidos hay tres posibles marcos de lectura. Por ejemplo, para la cadena 5' AATGCCCGCA 3' los tres posibles marcos de lectura son:

- 5' AAT GCC CGC A 3'
- 5' A ATG CCC GCA 3'
- 5' AA TGC CCG CA 3'

Si se iniciara en el cuarto nucleótido, su marco de lectura sería: 5' AAT GCC CGC A 3', que es el mismo que en primer supuesto.

Además, habría otros tres marcos de lectura para su cadena complementaria 3' TTACGGGCGT 5':

- 3' TTA CGG GCG T 5'
- 3' T TAC GGG CGT 5'
- 3' TT ACG GGC GT 5'

Por lo tanto, hay seis posibles marcos de lectura.

2.7.Alineamiento de secuencias

En bioinformática, el alineamiento de secuencias es una forma de representar y comparar dos o más secuencias para resaltar sus zonas de similitud, que podrían indicar relaciones funcionales o evolutivas entre los genes o proteínas consultados.

Si dos secuencias en un alineamiento comparten un ancestro común, las no coincidencias pueden interpretarse como mutaciones puntuales y los huecos como *indel* introducidos en uno o en ambos linajes.

CAPÍTULO 3

TECNOLOGÍAS EMPLEADAS

3. TECNOLOGÍAS EMPLEADAS

En este capítulo se expondrán las tecnologías y los lenguajes de programación que se han empleado para la realización del Trabajo de Fin de Grado.

3.1.VCF

En esta sección se explicará el formato de ficheros VCF, que contiene variaciones de secuencias genéticas.

3.1.1. Introducción

VCF [1], siglas de *Variant Call Format*, es un formato de ficheros de textos que especifica los cambios que hay que realizar sobre el genoma de la especie. En este caso, se trabaja con el genoma humano[5].

En esta sección se hará un breve repaso de la estructura del fichero VCF, aunque se puede encontrar información más detallada en su proyecto de GitHub[6].

[5] ftp://ftp.ncbi.nih.gov/genomes/Homo_sapiens/
[6] https://github.com/samtools/hts-specs

3.1.2. Composición

El fichero VCF contiene tres partes: las líneas que contienen la metainformación, la cabecera y los datos, que contienen la información sobre la posición en el genoma. Un pequeño ejemplo sería:

```
##fileformat=VCFv4.1                              (metainformación)
#CHROM POS ID REF ALT QUAL FILTER INFO            (cabecera)
20 3 . C G . PASS DP=100                         (datos)
20 2 . TC T . PASS DP=100                        (datos)
```

Figura 13: Ejemplo básico fichero VCF.

3.1.3. Metainformación

Se recomienda que las líneas de información que describen las entradas *INFO*, *FILTER* y *FORMAT* usadas en el cuerpo del fichero VCF estén incluidas en la sección de metainformación. Aunque son opcionales, si se incluyen, deben de estar escritas acorde a su estructura.

La metainformación se incluye después de la doble almohadilla (‘*##*’) y debe de ser de la siguiente forma:

##clave=valor

El campo *fileformat* es siempre necesario. Debe de estar en la primera línea del archivo y detalla la versión del formato VCF.

A continuación se especificará los otros campos que pueden ser incluidos en las líneas de metainformación, así como la estructura que deben de seguir:

- INFO: indica la estructura de la columna INFO y tiene el siguiente formato:

```
##INFO=<ID=ID,Number=number,Type=type,Description="desc
ription">
```

Figura 14: Campo INFO del fichero VCF

Donde *number* es un número entero que describe la cantidad de valores que pueden estar incluidos en el campo *INFO*. Si el campo tiene un valor por alelo alternativo, entonces este valor debería de ser 'A'; si el campo tiene un valor por cada uno de los posibles genotipos, entonces este valor debería de ser 'G'; si el número de valores varía, es desconocido o ilimitado, entonces su valor debería de ser el carácter punto ('.').

Los posibles valores del campo *type* son: *Integer, Float, Flag*, *Character* y *String*. *Flag* indica que el campo *INFO* no contiene una entrada valor, de forma que el valor de *number* debería de ser 0.

- FILTER: indica los filtros que han sido aplicados a la información tienen que tener el siguiente formato:

```
##FILTER=<ID=ID, Description="description">
```

Figura 15: Campo FILTER del fichero VCF.

- FORMAT: especifica el formato de los campos de genotipo y deberá de seguir el siguiente formato:

```
##FORMAT=<ID=ID,Number=number,Type=type,Description="
description">
```

Figura 16: Campo FORMAT del fichero VCF

- ALT: indica los alelos alternativos para las variaciones estructurales.

```
##ALT=<ID=type,Description=description>
```

Figura 17: Campo ALT del fichero VCF.

- ASSEMBLY: indica la localización del fichero FASTA que contiene los puntos de ruptura para las variaciones estructurales.

```
##assembly=url
```

Figura 18: Campo ASSEMBLY del fichero VCF.

- CONTIG: indica el cóntigo, formado por varios segmentos de ADN superpuestos que representan juntos una región consenso del ADN, usado en el fichero VCF.

```
##contig=<ID=ctg1,URL=dirección>
```

Figura 19: Campo CONTIG del fichero VCF

- SAMPLE: sirve para definir el mapeado del genoma, y ha de presentar la siguiente estructura:

```
##SAMPLE=<ID=S_ID,Genomes=G1_ID;G2_ID;
...;GK_ID,Mixture=N1;N2; ...;NK,Description=S1;S2; ...;SK>
```

Figura 20: Campo SAMPLE del fichero VCF.

- PEDIGREE: indica las relaciones entre los genomas.

```
##PEDIGREE=<Name_0=G0-ID,Name_1=G1-
ID,...,Name_N=GN-ID>
```

Figura 21: Campo PEDIGREE del fichero VCF.

También se puede relacionar a una base de datos:

```
##pedigreeDB=<url>
```

Figura 22: Relacionar base de datos con en el fichero VCF

3.1.4. Cabecera

La línea de la cabecera está formada como mínimo por ocho columnas delimitadas por el carácter tabulador ('\t'). Las ocho columnas que tienen que aparecer obligatoriamente por entrada son las siguientes:

- CHROM: especifica el cromosoma al que se aplicará las variaciones.
- POS: indica la posición de la variación, estando la primera base en la posición 1. Las posiciones están ordenadas numéricamente en sentido ascendente, es decir, las posiciones más bajas se encuentran al principio del fichero, dentro de cada secuencia de referencia de *CHROM*. Se permite tener múltiples entradas con la misma posición. Los telómeros se indican mediante el uso de la posición 0 o N+1, donde N es la longitud correspondiente al cromosoma o cóntigo.
- ID: un identificador único para cada variación. Si es una variación dbSNP (Single Nucleotide Polymorphism Database) se recomienda usar el número rs. dbSNP es un archivo público y libre de variaciones genéticas de diferentes especies

desarrollado y alojado por NCBI. Su objetivo es actuar como una única base de datos para contener todas las variaciones genéticas identificadas para ayudar a los investigadores. El número rs es un número de acceso utilizado por los investigadores y las bases de datos para referirse a un SNP específico.

No debería de haber más de un identificador por entrada. Si no hay un identificador disponible, entonces su valor será el carácter punto ('.').

- REF: identifica la base o bases de referencia. Cada base debe de ser A, C, G, T o N. Se puede incluir una o varias bases. El valor del campo *POS* se refiere a la posición de la primera base que aparece en el campo *REF*. Para inserciones o deleciones, el campo de *REF* y el de *ALT* deben de incluir la base anterior al evento (que debe de estar reflejado en el campo *POS*). Si la inserción o deleción es en la posición 1, hay que incluir la base posterior al evento.

- ALT: identifica la base o bases alternativas. Cada base debe de ser A, C, G, T o N. Se puede incluir una o varias bases. Si se quiere eliminar la base o bases del campo *REF*, entonces el valor del campo *ALT* tiene que ser el carácter punto ('.').

- QUAL: este parámetro indica la probabilidad de que el nucleótido o nucleótidos secuenciados sean erróneos según la escala de *Phred*, es decir, la probabilidad de que la base en la secuencia sea incorrecta. Ésta se define como:

$$\textit{Phred}\ \text{score} = -10 \log (\text{probabilidad de error})$$

Un valor alto significa que el SNP es probablemente real.

Por lo tanto, si la puntuación de *Phred* es de 10, la probabilidad de error de lectura en esta base es de 1/10*100, es decir, del 10% y el porcentaje de precisión sería del 90%. Si la puntuación de *Phred* fuera de 20, la probabilidad de error sería de 1/100*100, es decir, del 1% y la precisión del 99%. Si fuera 30, entonces la probabilidad de error sería del 0,1% y la precisión del 99,9%.

Este valor es usado frecuentemente para comparar la fiabilidad de los diferentes métodos de secuenciación.

- FILTER: indica que ha superado con éxito el filtro aplicado si el valor del campo es '*PASS*', si no se le ha aplicado ningún filtro entonces el valor del campo es el carácter punto ('.') y si no ha superado el filtro aplicado entonces el valor del campo es cualquier otro valor distinto a '*PASS*' y al carácter punto ('.').

- INFO: en esta columna se ubica la información adicional sobre las variantes del fichero VCF. Una etiqueta está formada por un par de clave-valor (*key-data*). Cada etiqueta debe de tener la siguiente estructura:

  key=data[,data]

  Las etiquetas deben de estar delimitadas por el carácter punto y coma (';'). Las *keys* arbitrarias están permitidas, aunque las siguientes están reservadas:

  - AA: indica el alelo ancestral.

- AC: especifica el número de veces que cada alelo *ALT* es representado.
- AF: es la frecuencia de cada alelo *ALT*.
- AN: este valor corresponde al número total de alelos en los genotipos.
- BQ: indica la media cuadrática de la calidad de la base en esta posición.
- CIGAR: describe cómo se alinea un alelo alternativo a un alelo de referencia.
- DB: indica qué variante del dbSNP es.
- DP: es el número de lecturas que pasaron el test de control de calidad interno.
- END: especifica la posición final de la variante descrita en esta entrada.
- H2: indica que el SNP se encuentra en el Hapmap 2.
- H3: indica que el SNP se encuentra en el Hapmap 3.
- MQ: indica la media cuadrática de la calidad del mapeo.
- MQ0: es el número de MAPQ.
- NS: describe el número de muestras con datos.
- SOMATIC: indica que la entrada es una mutación somática.

- 1000G: indica si pertenece a 1000 Genomes.

El formato exacto del subcampo *INFO* debe de estar especificado en metainformación, como se ha descrito anteriormente. Además, no es necesario incluir aquellas propiedades que no se posea de entre las que están reservadas.

A parte de las ochos columnas anteriores, también puede aparecer el campo '*Genotype*'. Si hay información del genotipo, los mismos tipos de datos deben de estar presentes en todas las muestras. Además, debe de aparecer primero el campo *FORMAT* que especifica los tipos de datos y su orden. Este campo *FORMAT* es un *string* alfanumérico separado por el carácter coma (','). Después de este campo *FORMAT*, aparece un único campo por muestra que tiene la información separada por el carácter coma (','). Este primer subcampo siempre tiene que tener el genotipo (*GT*) si está presente. Los demás subcampos no han de aparecer obligatoriamente. Las siguientes palabras están reservadas:

- GT: indica el genotipo de la muestra. Para un organismo diploide, indica los dos alelos que tiene la muestra. Los codifica de la siguiente manera: 0 para el alelo del campo *REF*, 1 para el primer alelo del campo *ALT*, 2 para el segundo alelo del campo *ALT*, y así sucesivamente. Cuando hay un único alelo, entonces la codificación sería:
    - 0/0: la muestra es homocigota de referencia.

    - 0/1: la muestra es heterocigoto, teniendo una copia de los alelos de los campos *REF* y de *ALT*.
    - 1/1: la muestra es homocigoto alternativo.

- DP: indica el valor de la *profundidad*, *depth*, que es el número de veces que un nucleótido es leído durante el proceso de secuenciación.
- FT: indica si ha pasado o no el filtro. '*PASS*' significa que ha pasado todos los filtros y el carácter punto ('.') indica que no se ha aplicado ningún filtro. Los demás valores tienen que ser descritos en metainformación.
- GL: indica la probabilidad del genotipo.
- GLE: indica la probabilidad, en la escala de *Phred*, de la ploidía, que es el número de juegos completos de cromosomas.
- PL: proporciona la probabilidad, en la escala de *Phred*, del genotipo redondeado al entero más próximo.
- GP: indica la probabilidad a posteriori del genotipo.
- CQ: indica la calidad del genotipo condicionado, que es un genotipo que depende de otros factores.
- HQ: indica la calidad del haploide.
- PS: indica el conjunto de fase, que es un conjunto de genotipos en fase a la que pertenece este genotipo.

- PQ: indica la probabilidad, en la escala de *Phred*, de que los alelos están ordenados incorrectamente.

- EC: es una lista de los alelos alternativos esperados para cada alelo alternativo, valga la redundancia, en el mismo orden que aparecen en el campo *ALT*.

- MQ: indica la media cuadrática de la calidad del mapeo.

Si alguna entrada no tiene un valor en alguno de estos campos o subcampos, entonces se tiene que poner el carácter punto ('.').

3.1.5. Datos

Cada línea representa una única variación y sus propiedades vienen reflejadas en las columnas. Los datos, igual que en la cabecera, van separados por el carácter tabulador ('\t').

3.2. FASTA

El formato FASTA [2] es un formato de fichero sencillo utilizado para representar las secuencias de nucleótidos o de proteínas. Los pares de bases o de aminoácidos se representan usando códigos de una única letra. Las líneas en blanco no están permitidas.

El fichero tiene una línea de cabecera que empieza por el carácter mayor que ('>'). Después de este carácter, puede aparecer opcionalmente el identificador de la secuencia y una descripción. Si se llegan a poner, entonces la palabra siguiente al carácter mayor que ('>') sería el identificador de la secuencia y el resto de la línea sería la descripción.

Después de la línea de cabecera, se encuentran las líneas de datos de la secuencia. En estas líneas se pueden escribir el carácter de tabulador ('\t'), el

carácter de salto de línea (‘*n*’) o el carácter espacio (‘␣‘) ya que será ignorado por cualquier programa que analice el fichero FASTA.

Dentro del mismo fichero puede haber una o varias secuencias debido a que éstas son leídas hasta que se encuentra el fin de fichero o el carácter mayor que (‘*>*’), que indicaría que hay una nueva secuencia.

Es recomendable que todas las líneas de texto tengan menos de 80 caracteres de longitud.

Se espera que las secuencias se representen en los códigos estándar IUB/IUPAC (Unión Internacional de Bioquímica/Unión Internacional de Química Pura y Aplicada International, en inglés, *Union of Biochemistry/International Union of Pure and Applied Chemistry*) para ácidos nucleicos y aminoácidos con las siguientes excepciones: las letras minúsculas son aceptadas y convertidas a letras mayúsculas y un único guión puede usarse para representar un hueco.

Los códigos admitidos para los ácidos nucleicos se muestran en la Tabla 1.

Código de ácido nucleico	Significado
A	A (adenina)
C	C (citosina)
G	G (guanina)
T	T (timina)
U	U (uracilo)
R	G o A (purina)
Y	T o C (pirimidina)
K	G o T (bases que son centona)
M	A o C (bases con grupo amino)
S	G o C (interacción fuerte)
W	A o T (interacción débil)

B	G, T o C
D	G, A o T
H	A, C o T
V	G, C o A
N	A, G, C o T
-	Hueco de longitud indeterminada

Tabla 1: Ácidos nucleicos soportados en un fichero FASTA.

Los códigos de aminoácidos admitidos aparecen reflejados en Tabla 2.

A	Alanina
B	Ácido aspártico
C	Cisteína
D	Ácido aspártico
E	Ácido glutámico
F	Fenilalanina
G	Glicina
H	Histidina
I	Isoleucina
K	Lisina
L	Leucina
M	Metionina
N	Asparagina
P	Prolina
Q	Glutamina
R	Arginina
S	Serina
T	Treonina
U	Selenocisteína
V	Valina

W	Triptófano
Y	Tirosina
Z	Ácido glutámico
X	Cualquiera
*	Parada de traducción
-	Hueco de longitud indeterminada

Tabla 2: Aminoácidos soportados en un fichero FASTA.

3.3.vcf-consensus

El script vcf-consensus[7] es un módulo de la herramienta VCFtools[8], consistente en un paquete de programas para trabajar con ficheros VCF. El objetivo de VCFtools es proporcionar métodos para facilitar el trabajo con ficheros VCF, que contienen datos de variación genética.

El script vcf-consensus sirve para aplicar las variaciones de un fichero VCF a un fichero FASTA para crear una secuencia consenso.

```
cat ref.fa | vcf-consensus file.vcf.gz > out.fa
```

Tabla 3: vcf-consensus

Una secuencia consenso especifica cuáles son los elementos comunes a las secuencias encontradas. Es decir, la secuencia consenso es la secuencia más probable. Si hay varias secuencias probables, entonces la secuencia consenso es una secuencia intermedia.

[7] http://vcftools.sourceforge.net/perl_module.html#vcf-consensus
[8] http://vcftools.sourceforge.net/

3.4. BLAST

En esta sección se comentará la potente y popular herramienta BLAST, que es muy utilizada por los investigadores de biología molecular.

3.4.1. Introducción

BLAST[9] [3] es un acrónimo de *Basic Local Alignment Search Tool* y se refiere a un conjunto de programas proporcionados por el NCBI para el alineamiento de secuencias de nucleótidos y de proteínas, así como para su comparación con secuencias de nucleótidos y de proteínas presentes en una base de datos seleccionada.

3.4.2. Entrada y salida

La entrada de BLAST es una secuencia, o secuencias, en formato FASTA o GenBank[10], que es la base de datos de disponibilidad pública de secuencias genéticas del NIH. La salida puede ser un HTML que es el formato que aparece por defecto, un texto plano, un texto en formato ASN.1 o un XML.

3.4.3. Características

Como se ha comentado anteriormente, BLAST alinea una secuencia problema, llamada *query* o *query sequence*, con las secuencias que se almacenan en una determinada base de datos. BLAST usa un algoritmo heurístico para encontrar las secuencias de la base de datos que tienen mayor parecido a la secuencia problema. Debido a que BLAST usa un algoritmo heurístico, no garantiza que las secuencias que alinea sean

[9] http://blast.ncbi.nlm.nih.gov/Blast.cgi
[10] http://www.ncbi.nlm.nih.gov/genbank/

homólogas y tengan la misma función biológica, simplemente provee posibles candidatos.

BLAST puede ser ejecutado remotamente en el servidor de NCBI o puede ser instalado localmente. Las ventajas de ejecutarlo remotamente en el servidor de NCBI son que el usuario no tiene que mantener actualizadas las bases de datos y que la búsqueda se hace en un clúster de computadoras, lo que otorga mayor rapidez al proceso. Sin embargo, las principales desventajas son que no se puede hacer búsquedas masivas debido a que es un recurso compartido, que no se puede personalizar las bases de datos en las que busca el programa y que no hay privacidad para las secuencias que son enviadas al servidor del NCBI debido a que no van cifradas. Estos inconvenientes se pueden resolver ejecutándolo localmente, aunque el usuario debe proporcionar sus propios recursos. En el caso de este proyecto, como se quiere ejecutar BLAST sobre una base de datos propia, se utiliza de forma local.

3.4.4. Familia BLAST

Como se ha comentado anteriormente, BLAST es una familia de programas. Los programas más usados son:

- blastn: compara una secuencia de nucleótidos con una base de datos que contenga también secuencia de nucleótidos.
- blastp: compara una secuencia proteica con una base de datos que contenga secuencias proteicas.
- blastx: la entrada es una secuencia de nucleótidos y compara sus seis posibles marcos de lectura con una base de datos de secuencias proteicas. Para ello, primero traduce la secuencia de

nucleótidos en sus seis posibles marcos de lectura y después las busca en la base de datos.

- tblastn: compara una sentencia proteica en una base de datos con los seis posibles marcos de lectura que tiene cada secuencia de nucleótidos de la base.

- tblastx: traduce la secuencia de nucleótidos en los seis posibles marcos de lectura que tiene y las compara con los seis posibles marcos de lectura que tiene cada secuencia de nucleótidos de la base de datos.

3.4.5. Distribución

BLAST es de dominio público y se puede descargar desde el servidor FTP de NCBI[11]. En este directorio se encuentran diferentes subdirectorios.

- db: se encuentran varias bases de datos BLAST en texto preformateado o en formato FASTA (en el subdirectorio /FASTA), así como sus respectivos MD5 para comprobar que se hayan descargado correctamente.

- demo: se localizan demostraciones de programas y documentación para los desarrolladores.

- documents: contiene la documentación de BLAST.

- executables: se hallan los ficheros binarios de las diferentes versiones que se han publicado de BLAST.

[11] ftp://ftp.ncbi.nlm.nih.gov/blast/

- matrices: se sitúan matrices de puntación de proteínas y nucleótidos.
- temp: es el directorio temporal con ficheros misceláneos.
- web_services: se ubica la documentación y el código de ejemplo para el servicio web de BLAST.

3.5.GEMINI

GEMINI[12], acrónimo de GEnome MINIng, es un framework flexible desarrollado por Umadevi Paila, Brad A. Chapman, Rory Kirchner y Aaron R. Quinlan para la exploración de la variación genética en el contexto de la gran cantidad de anotaciones del genoma disponibles para el genoma humano. GEMINI ofrece un sistema simple, flexible y potente para la exploración de las enfermedades genéticas y de la población. Debido a que su documentación es extensa, se resumirá la misma, aunque se puede acceder a los originales a través de su página web[13].

3.5.1. Funcionamiento

Primero se carga el fichero VCF (opcionalmente también puede usarse un fichero PED[14]) dentro de la base de datos. Cada variante es anotada automáticamente y comparada con varias anotaciones del genoma provenientes de diferentes fuentes, tales como ENCODE[15],

[12] http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1003153
[13] https://gemini.readthedocs.org/
[14] http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#ped
[15] https://www.encodeproject.org/

UCSC[16] , OMIM[17], dbSNP[18] , KEGG[19] y HPRD[20]. Toda esta información es almacenada en una base de datos portable SQLite[21] que permite explorar e interpretar las variantes genéticas. El sistema de gestión de la base de datos, SQLite, implementa la mayor parte del estándar SQL-92 del lenguaje SQL.[22]

Debido a que las variantes genéticas están en una base de datos SQLite, se puede utilizar el lenguaje SQL para realizar consultas[23].

[16] http://genome.ucsc.edu/
[17] http://omim.org/
[18] http://www.ncbi.nlm.nih.gov/projects/SNP/
[19] http://www.kegg.jp/
[20] http://www.hprd.org/
[21] http://www.sqlite.org/
[22] El esquema de la base de datos se puede encontrar en https://gemini.readthedocs.org/en/latest/content/database_schema.html
[23] https://gemini.readthedocs.org/en/latest/content/querying.html

CAPÍTULO 4

DISEÑO

4. DISEÑO

En este capítulo se explicará y se documentará la Especificación de Requisitos Software (ERS), que es una descripción completa del comportamiento del sistema desarrollado, del proyecto, y los casos de uso.

4.1. Especificación de Requisitos Software

4.1.1. Introducción

En esta sección se introducirá el propósito del sistema, su alcance, las definiciones, las referencias utilizadas y una visión general del mismo.

Esta Especificación de Requisitos Software ha sido elaborada teniendo en cuenta las características del sistema y la posibilidad de que sea mejorado en el futuro. Su estructura está basada en el estándar "*IEEE Recommended Practice For Software Requirements Specications*" (ANSI/IEEE std. 830, 1998) [6].

4.1.1.1. Propósito

En esta subsección se presentará la Especificación de Requisitos Software del proyecto. El sistema permitirá cargar

ficheros en formato VCF en una base de datos y, también, permitirá ejecutar la herramienta BLAST sobre este conjunto de datos.

El objetivo de esta especificación es definir de manera clara y precisa las funcionalidades y restricciones que tendrá el sistema que se va a desarrollar. Va dirigida al equipo de desarrollo de software y a las personas que harán uso del sistema terminado, por lo que se redactará en un lenguaje informal para que sea comprensible y se pueda entender.

4.1.1.2. Alcance del sistema

Se ha constatado la necesidad de crear una base de datos que almacene secuencias de nucleótidos y que pueda utilizar las diferentes posibilidades que permite la herramienta BLAST con este conjunto de datos. El objetivo de este proceso es desarrollar un sistema que permita realizar estas funciones con la finalidad de facilitar y optimizar el tiempo invertido por parte de los investigadores en realizar estas actividades.

4.1.1.3. Definiciones, siglas y abreviaciones

TÉRMINO	DESCRIPCIÓN
Sistema	Conjunto de hardware, software y usuarios.

Usuario	Persona encargada de aprovechar el sistema.
Nucleótido	Molécula formada por la unión de una base nitrogenada, una pentosa y un grupo fosfato. Desarrollado en la sección 2.1.
Secuencia genética	Sucesión de nucleótidos.
VCF	Abreviatura de Variant Call Format. Desarrollado en la sección 3.1.
FASTA	El formato FASTA. Desarrollado en la sección 3.2.
BLAST	La herramienta BLAST. Desarrollado en la sección 3.4.
GEMINI	El framework GEMINI. Desarrollado en la sección 3.5.

Tabla 4: Definiciones, siglas y abreviaturas de la Especificación de Requisitos Software

4.1.1.4. Referencias

REFERENCIAS
Software Engineering Standards Committee, IEEE Computer Society, «IEEE. IEEE/ANSI Std. 830: IEEE Recommended Practices for Software Requirements Specifications»

Tabla 5: Referencias: de la Especificación de Requisitos Software

4.1.1.5. Visión General

Esta subsección está dividida en tres partes:

1. Introducción: se proporciona una visión general de lo que es el apartado de la Especificación de Requisitos Software.

2. Descripción global: se detalla una descripción general del sistema para conocer sus funciones principales, los datos requeridos, y sus restricciones.

3. Requisitos: se enumeran los requisitos que el sistema tiene que cumplir y proporciona la información necesaria para su desarrollo.

4.1.2. Descripción global

En esta sección se describen los factores que afectan al producto y a sus requisitos.

4.1.2.1. Perspectiva del producto

El usuario interactuará con GEMINI para cargar los datos en la base de datos. Una vez cargados, podrá trabajar con ellos. El usuario también tendrá que interactuar para realizar búsquedas de secuencias genéticas en los datos de la base mediante la herramienta BLAST. Para que los datos almacenados en GEMINI puedan ser utilizados por BLAST, se utilizará la herramienta vcf-consensus.

Por lo tanto, el proceso depende principalmente del framework GEMINI, de vcf-consensus y de la herramienta BLAST.

4.1.2.2. Funciones del producto

Las funciones del producto son las siguientes:

- Creación de una base de datos que soporte datos de secuenciación genética.
- Cargar los datos del fichero VCF en la base de datos.
- Integración de la base de datos con la herramienta de alineamiento BLAST.

4.1.2.3. Características de los usuarios

El usuario será un investigador que deberá de tener unos conocimientos básicos de genética y de las herramientas GEMINI y BLAST. Además, también deberá de tener conocimientos de la lengua inglesa debido a que las herramientas utilizan ese idioma.

4.1.2.4. Restricciones

El sistema deberá de tener un diseño y una implementación sencilla debido a que va dirigido a usuarios con un conocimiento básico en informática.

4.1.2.5. Suposiciones y dependencias

Se ejecutará sólo en máquinas que tengan instalado un sistema operativo GNU/Linux.

Deberá de tener instalado GEMINI, BLAST, vcf-consensus (que se encuentra dentro de VCFtools), SQLite, junto a sus dependencias. Además, se hará uso de los comandos sed y cat. Estos dos comandos ya suelen venir instalados en casi todos los sistemas operativos GNU/Linux.

A continuación, se muestran las versiones de las diferentes herramientas utilizadas:

Software	Versión
GEMINI	0.9.1
BLAST	2.2.28
VCFtools	0.1.11
SQLite	3.8.2

sed	4.2.2
cat	8.2.1

Tabla 6: Número de versión de las herramientas utilizadas.

NOTA: los programas sed y cat normalmente ya vienen instalados en cualquier sistema operativo GNU/Linux.

Debido a la dependencia del sistema de estos softwares, si cambiaran de versión habría que comprobar si siguen soportando las características que tenían sus versiones anteriores.

4.1.3. Requisitos

El sistema se atendrá a los requisitos expresados en esta subsección. Todos los requisitos aquí expuestos son esenciales, es decir, de alta prioridad.

Los requisitos serán etiquetados con la letra *erre* mayúscula (‘*R*’) más un número natural único. Dos requisitos no pueden tener el mismo número.

Nº Requisito	Descripción
R1	El sistema deberá de ejecutarse en una máquina GNU/Linux.
R2	Crear la base de datos.

R3	Cargar los datos del fichero VCF en la base de datos.
R4	El usuario podrá trabajar con los datos de la base de datos a través del framework GEMINI.
R5	Exportar en un fichero VCF los datos seleccionados y que no hayan sido exportados anteriormente de la base de datos.
R6	Guardar en una base de datos los datos que se han exportado de la base de datos.
R7	Crear un fichero FASTA que se originará al aplicar el fichero VCF al fichero FASTA correspondiente mediante la herramienta vcf-consensus.
R8	Utilizar la herramienta BLAST en el fichero FASTA creado.
R9	El sistema irá mostrando mensajes sobre las operaciones que va

	realizando para que el usuario no tenga la sensación de que se ha bloqueado.
R10	El sistema deberá de ser diseñado para que su mantenimiento no sea complejo, de esta manera se facilitará la corrección de errores y la implantación de mejoras.

Tabla 7: Requisitos del sistema

4.1.3.1. Atributos del sistema

En esta subsección se detallan los atributos de calidad del sistema. Estos atributos son los siguientes:

- Fiabilidad.
- Mantenibilidad.
- Portabilidad.
- Seguridad.

4.1.3.1.1. Fiabilidad

En la especificación de los requisitos no se ha solicitado que el sistema tenga tolerancia de fallos. En consecuencia, no se ha contemplado en las fases de diseño y de implementación que el sistema siga funcionando sin perder ni funcionalidades ni prestaciones en caso de fallo. Aun así se han establecido diferentes comprobaciones a la

hora de utilizar los parámetros y se han programado mensajes de error para que el usuario sepa que está fallando el sistema.

En la especificación de los requisitos no se ha pedido que se diseñe el sistema para que soporte el trabajo simultáneo de varios usuarios. Por lo tanto, la concurrencia tampoco se ha contemplado en el diseño y en la implementación.

4.1.3.1.2. Mantenibilidad

El sistema está implementado y comentado siguiendo las buenas prácticas establecidas por la comunidad de desarrolladores. De esta forma, se facilita la comprensión y el mantenimiento del código.

4.1.3.1.3. Portabilidad

El sistema se tiene que ejecutar en sistemas operativos GNU/Linux. Por lo tanto, no se ha contemplado para el proyecto si las herramientas son multiplataforma o no, tan sólo que sean operativas en el sistema GNU/Linux.

4.1.3.1.4. Seguridad

La información que utiliza la herramienta no es sensible por lo que no hay que implementar ni un administrador de contraseñas, ni un sistema de control ni un sistema de registro de accesos al sistema. Como sólo hay un único rol, el investigador, no hay ni que crear ni que asignar roles a los usuarios.

4.2.Casos de uso

En esta sección se especificarán los casos de uso. Un caso de uso es una colección de escenarios de éxito y de fallo relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo. Los casos de uso permiten considerar y organizar los requisitos en el contexto de los objetivos, mejorando la comprensión y la cohesión.

4.2.1. Actores identificados

Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. El único actor en este proyecto es el investigador.

4.2.2. Casos de uso identificados

Los casos de uso identificados en formato breve son los siguientes:

- Carga de datos inicial: se cargan los datos procedentes del fichero VCF en la base de datos a través del framework GEMINI.
- Exportar datos: se exportan los datos especificados que se encuentran en la base de datos a un fichero VCF.
- Crear fichero FASTA: se crea un fichero FASTA a partir de la aplicación del fichero VCF a su correspondiente fichero FASTA.
- Utilizar la herramienta BLAST: se tiene que poder utilizar la herramienta BLAST con el nuevo fichero FASTA.

La descripción de los casos de uso en formato completo es la siguiente:

Nombre	Carga de datos inicial
Actor principal	El investigador
Personal involucrado e interés	El investigador, cuyo interés es tener en una base de datos los datos del fichero para poder trabajar con ellos.
Referencia a los requisitos	R1, R2, R3, R4, R9, R10
Precondiciones	El investigador tiene permisos de lectura del fichero VCF y, además, éste tiene los datos correctos.
Garantías de éxito	Los datos se han cargado satisfactoriamente en la base de datos.
Escenario principal de éxito	El investigador quiere tener los datos cargados en la base de datos. El investigador quiere trabajar con los datos cargados.
Extensiones	No hay.

Lista de tecnología y variaciones de datos	Un ordenador con un sistema operativo GNU/Linux y con el programa GEMINI, y sus dependencias.
Frecuencia	La frecuencia de la operación dependerá del investigador, algunos días cargará varios ficheros en la base de datos y otros días ninguno.

Tabla 8: Caso de uso I en formato completo

Nombre	Exportar datos
Actor principal	El investigador
Personal involucrado e interés	El investigador y su interés es obtener en un fichero VCF los datos seleccionados.
Referencia a los requisitos	R5, R6, R9, R10
Precondiciones	El usuario tiene permisos de lectura y escritura.

	Tiene que haber datos válidos almacenados en la base de datos.
Garantías de éxito	Creación de un fichero VCF con los datos seleccionados.
Escenario principal de éxito	El investigador necesita exportar los datos seleccionados a un fichero VCF.
Extensiones	No hay.
Lista de tecnología y variaciones de datos	Un ordenador con un sistema operativo GNU/Linux y con el programa GEMINI, y sus dependencias.
Frecuencia	La frecuencia de la operación dependerá del investigador, algunos días exportará varios ficheros y otros días ninguno.

Tabla 9: Caso de uso II en formato completo

Nombre	Crear fichero FASTA
Actor principal	El investigador

Personal involucrado e interés	El investigador, cuyo interés es la creación de un fichero FASTA que sea el resultado de aplicar un fichero VCF a su respectivo fichero FASTA.
Referencia a los requisitos	R7, R9, R10
Precondiciones	El usuario tiene que tener permisos de lectura y escritura. El fichero VCF tiene los datos correctos. El fichero FASTA al que se le aplicará el fichero VCF pertenece al cromosoma especificado en el fichero VCF.
Garantías de éxito	Un nuevo fichero FASTA, que será el resultado de aplicar el fichero VCF al fichero FASTA correspondiente.
Escenario principal de éxito	El investigador necesita aplicar el fichero VCF al fichero FASTA.
Extensiones	No hay.

Lista de tecnología y variaciones de datos	Un ordenador con un sistema operativo GNU/Linux y con la herramienta vcf-consensus y sus dependencias.
Frecuencia	La frecuencia de la operación dependerá del investigador, algunos días creará varios ficheros y otros días ninguno.

Tabla 10: Caso de uso III en formato completo

Nombre	Utilizar la herramienta BLAST
Actor principal	El investigador
Personal involucrado e interés	El investigador, cuyo interés es poder realizar búsquedas y alineaciones por similitud con la herramienta BLAST.
Referencia a los requisitos	R8, R9, R10
Precondiciones	El usuario debe de tener permisos de lectura y de escritura.

	El fichero FASTA debe de tener los datos correctos.
Garantías de éxito	El programa BLAST devuelve los resultados correctamente.
Escenario principal de éxito	El investigador quiere hacer búsquedas por similitud de secuencias de nucleótidos que se encuentran en el fichero FASTA.
Extensiones	No hay.
Lista de tecnología y variaciones de datos	Un ordenador con un sistema operativo GNU/Linux y con el programa BLAST, y sus dependencias.
Frecuencia	La frecuencia de la operación dependerá del investigador, algunos días la utilizará varias veces y otros días ninguna.

Tabla 11: Caso de uso IV en formato completo

CAPÍTULO 5

EJEMPLO DE USO

5. EJEMPLO DE USO

En este capítulo se describirá un ejemplo de uso. Debido al potencial de GEMINI y de BLAST, el número de ejemplos de uso es bastante alto y pueden ser bastantes complejos. Aquí se detallará un caso de uso sencillo pero completo, para facilitar la comprensión.

El capítulo está subdividido en cuatro secciones, correspondientes cada uno a cada caso de uso descrito en la sección 4.2.2.

5.1.1. Carga de ficheros VCF

EL primer paso de este proceso es cargar el fichero VCF en la base de datos[24].

Como se ha comentado en la subsección 3.1 dedicado a los ficheros VCF, los ficheros VCF tienen que tener obligatoriamente ocho columnas (*CHROM*, *POS*, *ID*, *REF*, *ALT*, *QUAL*, *FILTER*, *INFO*). La importación del fichero VCF a la base de datos mediante el framework GEMINI se realiza mediante el siguiente comando:

[24] https://gemini.readthedocs.org/en/latest/content/loading.html

```
$ gemini load –v [fichero_vcf] [baseDeDatos]
```

Tabla 12: Carga fichero VCF.

Se pueden utilizar varios núcleos para realizar la carga de una forma más rápida. Para ello, hay que utilizar el parámetro:

--cores [númeroDeNúcleo]

Adicionalmente, si se quiere utilizar la anotación VEP, la carga sería:

```
$ gemini load -v [fichero_vcf] -t VEP [baseDeDatos]
```

Tabla 13: Carga fichero VCF con anotación VEP

O si se quiere utilizar la notación snpEff:

```
$ gemini load -v [fichero_vcf] -t snpEff [baseDeDatos]
```

Tabla 14: Carga fichero VCF con anotación VEP

Como se ha comentado en la subsección 3.5 de GEMINI, éste soporta ficheros PED para establecer las relaciones familiares y la información fenotípica de las muestras del fichero VCF.

```
$ gemini load -v [fichero_vcf] -p [fichero_ped] -t snpEff [baseDeDatos]
```

Tabla 15: Carga fichero VCF con fichero PED.

Una vez que los datos han sido importados, el investigador ya puede trabajar con ellos, por ejemplo, buscando variantes[25].

[25] https://gemini.readthedocs.org/en/latest/content/querying.html

```
$ gemini query -q "[query]" [baseDeDatos]
```

Tabla 16: Consulta a la base de datos.

5.1.2. Exportación de los datos

Si el investigador quiere utilizar la herramienta BLAST con los datos que tiene almacenados, hay que aplicar el fichero VCF al fichero FASTA del cromosoma correspondiente.

Como se quiere evitar exportar dos veces la misma entrada, se lleva un registro de las entradas exportadas. Para poder llevarlo a cabo, se ha creado una tabla SQLite auxiliar en la que se guardará el id de la tabla Variants de la base de datos donde se encuentran los datos y una entrada que hará de booleano para saber si ya ha sido exportado o no.

En los ejemplos mostrados a continuación, la base de datos prueba.vcf hace referencia a la base de datos que se ha importado desde el fichero VCF, mientras que la base de datos control2.db es la base de datos auxiliar, donde se lleva a cabo el control de las entradas que ya se han exportado.

La creación de esta nueva base de datos con la tabla sería de la siguiente manera:

```
$ sqlite3 control2.db "CREATE TABLE tabla2 (id INTEGER, idVariants INTEGER NOT NULL, exportar BOOLEAN NOT NULL CHECK (exportar IN (0,1)) DEFAULT 0, PRIMARY KEY (ID));"
```

Tabla 17: Creación de la base de datos

Es decir, hemos creado una nueva base de datos (auxiliar.db) que contiene una tabla (tabla2) con las columnas id, que es un entero,

idVariants, otro entero que no puede ser NULL, y exportar, que es el booleano comentado anteriormente. Los valores de la columna exportar sólo pueden ser 0 o 1, siendo 0 que no se ha exportado y 1 que sí. Por defecto, el valor será 0.

A continuación, se adjunta la base de datos con los datos mediante el comando ATTACH DATABASE para poder trabajar con ella.

```
ATTACH DATABASE 'prueba.db' as C1;

ATTACH DATABASE 'control2.db' as C2;
```

Tabla 18: Se adjuntan las bases de datos

A continuación se hace un JOIN para seleccionar los variants_id que no se encuentran en la base de datos auxiliar (control2.db) y se insertan en la tabla tabla2 de control2.db.

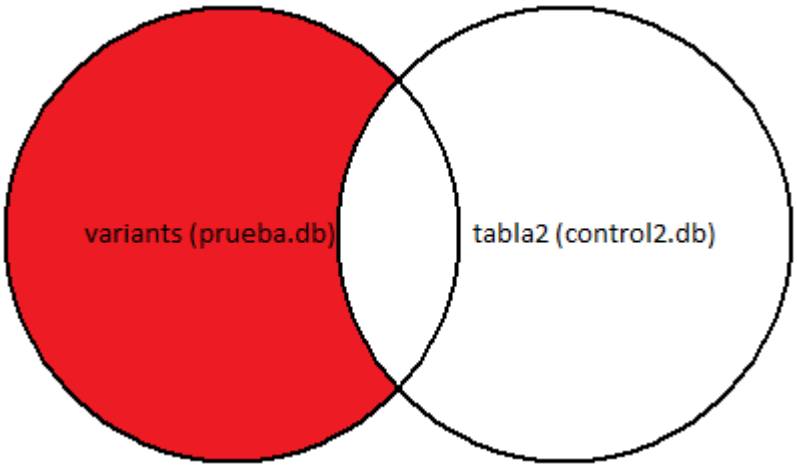

Figura 23: Left Excluding JOIN

En la Figura 23, el color rojo indica los datos que son seleccionadas al hacer el Left Excluding JOIN: todos los datos que se encuentran en la tabla variants pero no en la tabla tabla2. En concreto, esta query devuelve todos los registros de la columna variant_id de la tabla variants de la base

de datos prueba.db que no se encuentren en la columna idVariants de la tabla tabla2 de la base de datos control2.db. Es decir, estamos seleccionado todos aquellas variants_id que no se encuentran en nuestra tabla auxiliar y no se han exportado.

El resultado que devuelve esta query lo guardamos en la columna idVariants de la tabla2. Se recuerda que en SQLite, todo campo Integer que no se especifica al insertar se autoincrementa. Además, en la definición de la creación de la tabla, se ha especificado que la columna exportar era por defecto 0. Este es el motivo por el cual no se especifican ambas columnas al insertar.

```
insert into C2.tabla2 (idVariants) SELECT a.variant_id FROM C1.variants a
LEFT JOIN C2.tabla2 b ON a.variant_id=b.idVariants WHERE b.idVariants
IS NULL;
```

Tabla 19: Inserción y realización del LEFT JOIN

Después de haber insertado los id, se seleccionan los datos para construir el fichero VCF, redireccionando antes la salida de este resultado a un nuevo fichero, en los siguientes ejemplos se le denomina salida.vcf.

```
SELECT substr(a.chrom,4) AS "#CHROM",a.start+1 AS POS,
COALESCE(a.vcf_id,".") AS ID, a.ref AS REF, COALESCE(a.alt,".") AS
ALT, COALESCE(a.qual,".") AS QUAL, COALESCE(a.filter,".") AS
FILTER, a.info AS INFO FROM C1.variants a, C2.tabla2 b WHERE
a.variant_id = b.idVariants AND b.exportar=0;
```

Tabla 20: Construcción fichero VCF.

Ahora queda actualizar los registros exportados cambiando el campo exportar a 1 para no volver a exportarlos otra vez.

Por último, queda escribir la cabecera en el fichero VCF generado. Esto se realiza mediante el comando sed[26].

```
$ sed -i "1i ##fileformat=VCFv4.1" salida.vcf
```

Tabla 21: Comando sed.

El parámetro *–i* indica que el cambio se realice en el fichero. Con el flag *1i* se indica que se inserte la línea de texto en la primera línea del fichero.

5.1.3. Obtención del fichero FASTA

Una vez exportados los datos en un fichero VCF, el siguiente paso es aplicar este fichero VCF a su correspondiente fichero FASTA. Para ello se utiliza la herramienta vcf-consensus.

Este script requiere que el fichero VCF sea comprimido con bgzip e indexado con tabix[27]. El fichero VCF se comprime y se indexa con los siguientes comandos.

```
$ bgzip salida.vcf

$ tabix -p vcf salida.vcf.gz
```

Tabla 22: Pasos previos a la utilización de vcf-consensus.

Una vez realizados los pasos previos anteriores, ya se puede utilizar el script vcf-consensus para aplicar el fichero VCF al fichero FASTA correspondiente.

[26] http://manpages.ubuntu.com/manpages/hardy/man1/sed.1.html
[27] Estas dos herramientas ya vienen incluidas al instalar vcf-consensus

```
$ cat ficheroFasta.fasta | vcf-consensus salida.vcf.gz > salida.fa
```

Tabla 23: Aplicación del fichero VCF al fichero FASTA.

5.1.4. Utilización de la herramienta BLAST

Una vez se tiene el fichero salida.fa con los datos de la base de datos, ya se pueden realizar búsquedas y alineaciones de secuencias de nucleótidos con la herramienta BLAST.

Normalmente se quiere comparar una misma secuencia con varios ficheros. Para ello se reúnen todos los ficheros en uno único.

El programa makeblastdb generar bases de datos de BLAST a partir de ficheros FASTA. El texto de la línea de la definición se almacena en la base de datos de BLAST y se muestra en el informe generado por BLAST, pero no es posible buscar secuencias individuales usando blastdbcmd o limitar la búsqueda con la opción *–seqidlist*. El flag *–parse_seqids* indica a makeblastdb que puede recuperar las secuencias basándose en los identificadores de secuencia. En este caso, cada secuencia debe de tener un identificador único. El identificador, que se encuentra en la línea de la definición, debe de empezar con el carácter mayor que (‘>’) y no puede contener espacios.

La estructura recomendable de los identificadores (ID) de los ficheros FASTA está especificada en Tabla 28: Formato del ID del fichero FASTA.

A continuación se muestra un fichero con dos secuencias juntas.

```
$ cat salida.fa

>lcl|7 secuencia 1
```

ACAGAAAAGGGACCTCACATTCTGTATTTGTCCCGATTGGCTAG
CAACTTAGAACTTTTTTAAAGAGGCCCAGGCAGAGGAGAACAAAG
GAAGGAGGAAGTAACTTGTGGAATGCTGAAAAAAGTAAAAACAC
CTTCG

>lcl|7g secuencia 2
ACAGAAAAGGGACCTCACATTCTGTATTTGTCCCGATTGGCTAG
CAACTTAGAACTTTATAGAGGCCCAGGCAGAGGAGAACAAAGGA
AGGAGGAAGTAACTTGTGGAATGCTGAA

Tabla 24: Fichero FASTA

La secuencia de nucleótidos que se va a buscar es la siguiente:

$ cat salidaIncompleta.fa

>7
ACAGAAAAGGGACCTCACATTCTGTA

Tabla 25: Secuencia a buscar.

Dicha secuencia se encuentra en las dos anteriores secuencias (destacado en negrita).

Se crea la base de datos BLAST con el programa makeblastdb. El parámetro *–in* indica el fichero que contiene la secuencia entrada. El parámetro *–parse_seqids* indica que lea e indexe el ID de las secuencias. El parámetro *–dbtype* indica el tipo de la base de datos, en este caso es de nucleótidos. Y el parámetro *–out* indica que la base de datos se va a guardar en el fichero dosDB.

```
$ makeblastdb -in salida.fa -parse_seqids -dbtype nucl -out dosDB

Building a new DB, current time: 01/03/2015 23:49:11
New DB name:   dosDB
New DB title:  salida.fa
Sequence type: Nucleotide
Keep Linkouts: T
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 2 sequences in 0.000438929 seconds.
```

Tabla 26: Creación de la base de datos con el comando makeblastdb.

Ahora se puede realizar la búsqueda. Como se va a comparar una secuencia de nucleótidos con una base de datos formada por nucleótidos, se utiliza el programa blastn.

```
$ blastn -db dosDB -evalue 1.0E-10 -word_size 4 -query salidaIncompleta.fa
BLASTN 2.2.28+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.

Database: salida.fa
        2 sequences; 254 total letters
Query= 7

Length=26
                                                    Score     E
Sequences producing significant alignments:                (Bits)  Value

lcl|7g  secuencia 2                                 49.1    8e-12
lcl|7   secuencia 1                                 49.1    8e-12

>lcl|7g secuencia 2
Length=116

 Score = 49.1 bits (26),  Expect = 8e-12
 Identities = 26/26 (100%), Gaps = 0/26 (0%)
 Strand=Plus/Plus

Query  1   ACAGAAAAGGGACCTCACATTCTGTA  26
           ||||||||||||||||||||||||||
Sbjct  1   ACAGAAAAGGGACCTCACATTCTGTA  26
```

```
>lcl|7  secuencia 1
Length=138

 Score = 49.1 bits (26),  Expect = 8e-12
 Identities = 26/26 (100%), Gaps = 0/26 (0%)
 Strand=Plus/Plus

Query  1   ACAGAAAAGGGACCTCACATTCTGTA  26
           ||||||||||||||||||||||||||
Sbjct  1   ACAGAAAAGGGACCTCACATTCTGTA  26

Lambda      K        H
   1.33    0.621     1.12

Gapped
Lambda      K        H
   1.28    0.460    0.850

Effective search space used: 4840

  Database: salida.fa
    Posted date:  Jan 3, 2015  11:49 PM
  Number of letters in database: 254
  Number of sequences in database:  2

Matrix: blastn matrix 1 -2
Gap Penalties: Existence: 0, Extension: 2.5
```

Tabla 27: Búsqueda con BLAST

CAPÍTULO 6

CONCLUSIONES Y LÍNEAS FUTURAS

6. CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se expondrá las conclusiones a las que se ha llegado después de realizar este este Trabajo de Fin de Grado.

6.1.Conclusiones

El objetivo del trabajo es el diseño y la implementación de una base de datos que almacene secuencias de nucleótidos y que puedan ser utilizadas por la herramienta BLAST. Los objetivos del proyecto se han cumplido satisfactoriamente.

Hoy en día se puede identificar las variaciones genéticas entre varios genotipos humanos gracias a los avances en la tecnología de secuenciación del ADN. Sin embargo, el reconocer qué variantes producen enfermedades es todavía un reto no alcanzado.

GEMINI integra la variación genética con un conjunto diverso y adaptable de anotaciones del genoma en una base de datos unificada para facilitar la interpretación y la exploración de los datos almacenados. Gracias a la integración de los datos en una base de datos, GEMINI permite la realización de consultas complejas, así como estudios de genética en familias humanas.

Estas características hacen que GEMINI se convierta en un poderoso y flexible framework para trabajar con las variaciones genéticas en humanos.

Además, a pesar de que GEMINI es un framework reciente, del año 2013, tiene una comunidad activa[28] con usuarios experimentados que resuelven las dudas que se puedan tener respecto del uso del mismo.

BLAST es un conjunto de programas potente que permite la alineación de secuencias de nucleótidos y de proteínas. Así, por ejemplo, si se detecta una nueva secuencia, se puede buscar con la herramienta BLAST para compararla con otras secuencias almacenadas y facilitar la investigación sobra la función biológica que pueda tener dicha secuencia.

BLAST es un software muy conocido y muy usado por la comunidad científica para la búsqueda y la alineación de secuencias genéticas gracias a su mayor rapidez, a costa de un poco menos de precisión, que el algoritmo Smith-Waterman[29]. La velocidad es un factor importante a tener en cuenta debido a que cada vez hay disponible más secuencias con las que comparar. Además, permite realizar búsquedas en diferentes bases de datos y en diferentes genomas.

BLAST tiene una gran cantidad de documentación online y hay diferentes libros publicados que tratan sobre él. Además, hay tutoriales y dudas resueltas de otras personas que están fácilmente accesibles en la red.

En consecuencia, la utilización de ambos programas es útil porque se complementan una con la otra, aportando al investigador una nueva herramienta más sencilla que facilitará su labor.

[28] https://groups.google.com/forum/#!forum/gemini-variation
[29] T. F. Smith, M. S. Waterman, “Identification of Common Molecular Subsequences,” Journal of Molecular Biology, 147(1): pp: 195-197, 1981

Este proyecto es una muestra más de la importancia que tiene la rama de la bioinformática para el avance del estudio de la genética.

Por lo tanto, la bioinformática provee de herramientas que ahorran tiempo, esfuerzo y costes. Sin estas herramientas, al investigador no le quedaría otra opción más que realizar ciertas tareas de forma manual.

La bioinformática, al igual que la informática, es una ciencia relativamente joven, si se comparan con otras ciencias como las matemáticas, la física o la química. Este campo está teniendo grandes avances debido a su dependencia de la informática, la cual también está teniendo un gran crecimiento. Muestra de este crecimiento son los avances y la optimización de los procesadores o de las unidades de procesamiento gráfico. Gracias a estos avances en la informática, otras ciencias, que dependen de ella, también avanzan ya que permite la realización y el procesamiento de cálculos y operaciones más complejas de manera más rápida y eficaz.

6.2. Líneas futuras de trabajo

En esta sección se propondrán las posibles mejoras del proyecto, el cual está dirigido para secuencias genéticas de humanos.

Se pueden realizar muchas mejoras, el límite es la imaginación y las necesidades reales del sistema, no sirve de nada implementar una nueva función si después no se va a usar. A continuación se explicará algunas posibles mejoras, aunque como se ha comentado anteriormente, de nada sirve añadir una nueva función si no se utiliza.

- Una mejora sería soportar secuencias genéticas de otras especies. Este cambio afectaría principalmente a GEMINI, que es un framework dirigido al genoma humano. Con BLAST no habría

problema porque acepta secuencias genéticas de cualquier especie.

- Otra mejora sería soportar secuencias de aminoácidos, que forman las proteínas. Nuevamente, este cambio afectaría a GEMINI pero no a BLAST, ya que soporta esta opción.

- También podría mejorarse procurando elaborar un sistema multiplataforma ya que este proyecto se ha diseñado solamente para GNU/Linux, pudiendo ser un inconveniente para aquellos investigadores que no tengan acceso a este sistema operativo.

6.3. Dificultades encontradas

El principal inconveniente encontrado ha sido la falta de conocimiento inicial en la materia de genética. Las herramientas utilizadas tienen como usuario final a investigadores, que son expertos en la materia. Por lo tanto, usan conocimientos propios que son desconocidos para una persona que nunca ha estudiado este campo científico con tanta especificidad. Además, hay que sumarle el amplio uso de tecnicismos en esta rama de la biología, lo que hace que sea aún más compleja la comprensión de determinados contenidos. Por otra parte, la mayoría de fuentes de información que se han manejado para la elaboración del trabajo se presentaban en inglés, lo que también ha supuesto una dificultad añadida.

6.4. Conocimientos adquiridos

La realización de este Proyecto de Fin de Grado ha supuesto el descubrimiento, a título personal, de una rama de la ciencia en la que convergen la ingeniería informática y la biología, concretamente en el campo de la genética.

Durante el Grado el Ingeniería Informática, sólo se estudian unas pocas ramas, y no en demasiada profundidad, de la informática: base de datos, comunicación y seguridad, lenguajes de programación, ingeniería del software, inteligencia artificial y sistemas. Sin embargo, hay otras ramas, como la Computación Científica, de la que la bioinformática forma parte de ella, que no se han visto ni mencionado.

Además, también se han ampliado y profundizado el uso de herramientas de las bases de datos y del lenguaje SQL. Estos conceptos han sido tratados brevemente de forma teórica en la carrera. Con este proyecto se ha podido poner en práctica y ampliar dichos conocimientos en un ambiente real de trabajo.

Lo mismo ocurre con el sistema GNU/Linux. Se ha tenido que profundizar en el conocimiento de las herramientas para poder instalarlas y configurarlas correctamente debido, principalmente, a sus dependencias de software.

6.5. Posibles utilidades

Este proyecto permite el estudio de algunas de las moléculas más relevantes de la vida: el ADN y el ARN.

Una aplicación de este proyecto sería la búsqueda de enfermedades genéticas. Por ejemplo, si se descube un nuevo gen causante de una enfermedad en una especie distinta al ser humano, como por ejemplo, en los ratones, el investigador podría buscar este gen en la secuencia humana para ver si los humanos portan este mismo gen.

Además permite la identificación y el alineamiento de las variaciones genéticas entre varias secuencias genéticas de humanos para encontrar genes que desempeñen funciones biológicas importantes.

También puede tener una aplicación en el mundo de la criminología, en tanto que el sistema permite cotejar dos secuencias de ADN, permitiendo la identificación de personas.

Otra posible utilidad del sistema sería el establecimiento de relaciones de parentesco entre dos personas, pudiéndose ser utilizado, por ejemplo, en pruebas de paternidad.

Por último, se podría introducir en la base de datos la secuencia genética responsable de alguna enfermedad. De forma que se puede usar como referencia para detectar dicha enfermedad en otros sujetos.

APÉNDICE A: FORMATO DEL ID

En este apéndice se especifica el formato recomendable que debe de tener el ID del fichero FASTA. No tiene que haber espacios en el ID[30].

Type	Format(s)	Example(s)
local	lcl|integer	lcl|123
	lcl|string	lcl|hmm271
GenInfo backbone seqid	bbs|integer	bbs|123
GenInfo backbone moltype	bbm|integer	bbm|123
GenInfo import ID	gim|integer	gim|123

[30] Tabla tomada de http://www.ncbi.nlm.nih.gov/toolkit/doc/book/ch_demo/#ch_demo.TF.1

GenBank	gb|accession|locus	gb|M73307|AGMA13GT
EMBL	emb|accession|locus	emb|CAM43271.1|
PIR	pir|accession|name	pir||G36364
SWISS-PROT	sp|accession|name	sp|P01013|OVAX_CHICK
patent	pat|country|patent|sequence	pat|US|RE33188|1
pre-grant patent	pgp|country|application-number|seq-number	pgp|EP|0238993|7
RefSeq	ref|accession|name	ref|NM_010450.1|
general database reference	gnl|database|integer	gnl|taxon|9606
	gnl|database|string	gnl|PID|e1632
GenInfo integrated database	gi|integer	gi|21434723
DDBJ	dbj|accession|locus	dbj|BAC85684.1|
PRF	prf|accession|name	prf||0806162C

PDB	pdb|entry|chain	pdb|1I4L|D
third-party GenBank	tpg|accession|name	tpg|BK003456|
third-party EMBL	tpe|accession|name	tpe|BN000123|
third-party DDBJ	tpd|accession|name	tpd|FAA00017|
TrEMBL	tr|accession|name	tr|Q90RT2|Q90RT2_9HIV1
genome pipeline	gpp|accession|name	gpp|GPC_123456789|
named annotation track	nat|accession|name	nat|AT_123456789.1|

Tabla 28: Formato del ID del fichero FASTA.

BIBLIOGRAFÍA

[1] "1000 genomes," [Online]. Available: http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41.

[2] "NCBI. Blast. Documentación formato FASTA," [Online]. Available: http://blast.ncbi.nlm.nih.gov/blastcgihelp.shtml.

[3] N. H. Bergman, Comparative Genomics Volume 1 y 2, Humana Press, 2007.

[4] T. Tao, T. Madden, C. Camacho and L. Szilagyi, "BLAST FTP Site," in *BLAST Help*, National Center for Biotechnology Information, 2008.

[5] "TIOBE Software," [Online]. Available: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

[6] "NCBI. Servidor FTP. Documentación BLAST," [Online]. Available: ftp://ftp.ncbi.nlm.nih.gov/blast/blastftp.txt.

[7] "HTML," [Online]. Available: http://www.w3.org/.

[8] U. Paila, B. A. Chapman, R. Kirchner and A. R. Quinlan, "GEMINI: Integrative Exploration of Genetic Variation and Genome Annotations. PLoS Comput Biol 9(7): e1003153. doi:10.1371/journal.pcbi.1003153," 2013. [Online]. Available: https://gemini.readthedocs.org/en/latest/.

[9] "Human Genome Project," [Online]. Available: http://www.genome.gov/10001772.

[10] "Variant Call Format," 7 Diciembre 2013. [Online]. Available: http://samtools.github.io/hts-specs/VCFv4.1.pdf.

[11] Software Engineering Standards Committee, IEEE Computer Society, "IEEE. IEEE/ANSI Std. 830: IEEE Recommended Practices for Software Requirements Specifications".

[12] I. Korf, M. Yandell and J. Bedell, Blast, O'Reilly Media, 2003.

[13] J. Alcamí, J. J. Bastero, B. Fernández, J. M. Gómez de Salazar, M. J. Méndez, A. Ogayar and M. Sánchez, Biología. Ciencias de la naturaleza y de la salud. 2 Bachillerato, 1ª ed., SM, 2003.

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Wed Jan 07 01:22:21 CET 2015
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)